

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

RECEIVED
CENTRAL FAX CENTER
SEP 16 2010

IN THE CLAIMS

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- Claim 1 (withdrawn) A method for detecting attempted intrusions in a database application, the method comprising:
- monitoring for an SQL statement, said SQL statement executable in said database application and intended to exploit a vulnerability;
 - actuating said SQL statement to discover an atomic SQL command;
 - analyzing said atomic SQL command against a pre-defined set of detection rules.
- Claim 2 (withdrawn) The method according to claim 1, wherein said vulnerability is a buffer overflow in a SQL procedure.
- Claim 3 (withdrawn) The method according to claim 1, wherein said vulnerability is a buffer overflow in a call from SQL to an operating system function.
- Claim 4 (withdrawn) The method according to claim 1, wherein said vulnerability is an attempt to escalate privileges of a user in said database application.
- Claim 5 (withdrawn) The method according to claim 1, wherein said vulnerability is an attempt to escalate privileges within an operating system.
- Claim 6 (withdrawn) The method according to claim 1, wherein said vulnerability is an attempt to insert an invasive SQL statement into a parameter of stored procedures.
- Claim 7 (withdrawn) A method for detecting an anomalous command in a database application, the method comprising:

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

RECEIVED
CENTRAL FAX CENTER
SEP 16 2010

actuating said database application in order to discover a form of a set of
authorized SQL statements and commands and to discover appropriate parameters for said
statements and commands;

generating a rule set of said discovered form of said authorized SQL statements;

monitoring for SQL statements executable in said database application which do
not match said generated rule set of forms of authorized SQL statements.

Claim 8 (withdrawn) The method according to claim 7, wherein said anomalous
command is a SELECT statement.

Claim 9 (withdrawn) The method according to claim 7, wherein said anomalous
command is an UPDATE statement.

Claim 10 (withdrawn) The method according to claim 7, wherein said anomalous
command is an INSERT statement.

Claim 11 (withdrawn) The method according to claim 7, wherein said anomalous
command is a DELETE statement.

Claim 12 (withdrawn) The method according to claim 7, wherein said anomalous
command is a call to a stored procedure.

Claim 13 (withdrawn) The method according to claim 7, wherein said anomalous
command is a batch script.

Claim 14 (withdrawn) A method for detecting attempts to access a database application
from invalid sources, the method comprising:

actuating said database application in order to discover a normal set of authorized
SQL sources;

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

generating a rule set of characteristics of connecting at least one of said normal set of SQL sources;

monitoring for SQL statements executable in said database application which do not match said generated rule set of valid forms for authorized SQL statements.

Claim 15 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on a location of an SQL source.

Claim 16 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on a network address of an SQL source.

Claim 17 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on a host name of an SQL source.

Claim 18 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on a domain name of an SQL source.

Claim 19 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on a time of activity of an SQL source.

Claim 20 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on an application name of an SQL source.

Claim 21 (withdrawn) The method according to claim 14, wherein a characteristic of said rule set is based on a behavior of an SQL source.

Claim 22 – 30 (cancelled)

Claim 31 (withdrawn) A method for detecting activity designed to breach security of a database application, the method comprising:

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

monitoring for discrete events executable in said database application and
intended to breach a security mechanism associated with said database application;
actuating each discrete database event;
analyzing said database events against a pre-defined set of detection rules.

Claim 32 (withdrawn) The method according to claim 31, wherein said activity is a brute-force guessing of usernames in said database application.

Claim 33 (withdrawn) The method according to claim 31, wherein said activity is the brute-force guessing of usernames and passwords for default accounts in said database application.

Claim 34 (withdrawn) The method according to claim 31, wherein said activity is the brute-force guessing of usernames and passwords for well-known accounts in said database application.

Claim 35 (withdrawn) The method according to claim 31, wherein said activity is the scripting of password guessing against the database application.

Claim 36 (withdrawn) A method for detecting suspicious activity in a database application, the method comprising:

monitoring for SQL statements executable in said database application which
contain characteristics indicative of an attack;
actuating each batch statement in order to discover atomic SQL commands;
analyzing said atomic SQL commands against a pre-defined set of rules to
identify said suspicious activity.

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

Claim 37 (withdrawn) The method according to claim 36, wherein said suspicious activity is a use of comments within an SQL statement.

Claim 38 (withdrawn) The method according to claim 36, wherein said suspicious activity is a use of a UNION keyword within an SQL statement.

Claim 39 (withdrawn) The method according to claim 36, wherein said suspicious activity is a use of a keyword designed to suppress auditing data.

Claim 40 (withdrawn) A method for detecting use of keywords to suppress auditing of attacks in a database application, the method comprising:

monitoring for SQL statements that contain a keyword, where said keyword results in audit data being suppressed;

detecting a suppressed SQL statement;

detecting a conclusion of said suppressed SQL statement;

determining that no execution of said keyword designed to suppress said SQL statement actually occurred.

Claim 41 (withdrawn) The method according to claim 40, further comprising a use of passwords designed to cause an auditing system to suppress text of said SQL statement and masking malicious activity.

Claim 42 (withdrawn) A host-based intrusion prevention method for blocking attacks on database applications, the method comprising:

detecting an attack occurring through a session with said database application;

identifying a source of said attack;

implementing a method of stopping said attack source;

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

implementing a method of preventing further attacks from said attack source.

Claim 43 (withdrawn) The method according to claim 42, wherein said method of stopping said attack source is killing a user connection of said attack source.

Claim 44 (withdrawn) The method according to claim 42, wherein said method of stopping said attack source is sending a reset to said attack source.

Claim 45 (withdrawn) The method according to claim 42, wherein said method of stopping said attack source is blocking a SQL command.

Claim 46 (withdrawn) The method according to claim 42, wherein said method of stopping said attack source is intercepting and filtering a SQL command.

Claim 47 (withdrawn) The method according to claim 42, wherein said method of stopping said attack source is throwing an exception.

Claim 48 (withdrawn) The method according to claim 42, wherein said method of preventing further attacks is disabling an account from being used.

Claim 49 (withdrawn) The method according to claim 42, wherein said method of preventing further attacks is killing any future attempts from said attack source.

Claim 50 (withdrawn) A method for detecting attempts to inject SQL into a database application, the method comprising:

monitoring for SQL statements executable in said database application and intended to run queries not designed to be run by a middle-tier application;

analyzing said SQL statement's identifying characteristics indicative of SQL injection;

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

implementing an action upon detection of identifying characteristics indicative of SQL injection.

Claim 51 (withdrawn) The method according to claim 50, wherein said action is causing a security alert to be fired.

Claim 52 (withdrawn) The method according to claim 50, wherein said action is causing the SQL statement to be blocked.

Claim 53 (withdrawn) A method for detecting attempts to inject SQL into a database application, the method comprising:

listening to SQL queries executable on said database application for a determined period of time;

tokenizing SQL statements into standard forms;

recording a combination and an order of tokens expected;

analyzing SQL statements received later to identify those that do not conform to said expected combination of tokens.

Claim 54 (withdrawn) A method for detecting malicious activity in a database application, the method comprising:

listening to SQL queries executable on said database application;

analyzing SQL statements by applying regular expressions to detect vulnerabilities;

sending alerts when an SQL statement matching a regular expression is discovered.

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

Claim 55 (withdrawn) The method according to claim 54, wherein said regular expression is designed to detect a buffer overflow in a call from SQL to a built-in database function.

Claim 56 (withdrawn) The method according to claim 54, wherein said regular expression is designed to detect a buffer overflow in a call from SQL to an operating system function.

Claim 57 (withdrawn) The method according to claim 54, wherein said regular expression is designed to detect an attempt to escalate privileges of a user in said database application.

Claim 58 (withdrawn) The method according to claim 54, wherein said regular expression is designed to detect an attempt to insert an SQL statement into a parameter of stored procedures.

Claim 59 (withdrawn) The method according to claim 54, wherein said regular expression is designed to detect an attempt to escalate privileges of a user in an operating system.

Claim 60 (withdrawn) A method for detecting activity which may result in cross-site scripting vulnerabilities, the method comprising:

monitoring for SQL statements executable in said database application;
actuating each batch statement in order to discover atomic SQL commands;
examining an atomic SQL command for HTML tags.

Claim 61 (withdrawn) The method according to claim 60, wherein said atomic SQL command contains an HTML tag.

Claim 62 (withdrawn) The method according to claim 61, wherein said HTML tag is unencoded.

Claim 63 (withdrawn) The method according to claim 61, wherein said HTML tag is hex encoded.

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

Claim 64 (withdrawn) A method for monitoring all activity for security auditing, the method comprising:

monitoring for an event generated by a database application;
actuating said event;
recording said event.

Claim 65 (withdrawn) The method according to claim 64, wherein said event being generated comprises an SQL statement.

Claim 66 (withdrawn) The method according to claim 64, wherein said event being generated comprises failed logins and successful logins.

Claim 67 (withdrawn) The method according to claim 64, wherein said event being generated comprises incomplete attempts to access said database application.

Claim 68 (withdrawn) The method according to claim 64, wherein said event being generated comprises DBA activity.

Claim 69 (withdrawn) The method according to claim 64, wherein said event being generated comprises changes to a configuration.

Claim 70 (withdrawn) The method according to claim 64, wherein said event being generated comprises enabling of application roles.

Claim 71 (withdrawn) The method according to claim 64, wherein said event being generated comprises a method of granting, revoking, or denying permissions or privileges.

Claim 72 (withdrawn) The method according to claim 64, wherein said event being generated comprises a utility event.

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

- Claim 73 (withdrawn) The method according to claim 72, wherein said utility event is a backup command.
- Claim 74 (withdrawn) The method according to claim 72, wherein said utility event is a restore command.
- Claim 75 (withdrawn) The method according to claim 72, wherein said utility event is a bulk insert command.
- Claim 76 (withdrawn) The method according to claim 72, wherein said utility event is a BCP command.
- Claim 77 (withdrawn) The method according to claim 72, wherein said utility event is a DBCC command.
- Claim 78 (withdrawn) The method according to claim 64, wherein said event being generated comprises a server shutdown.
- Claim 79 (withdrawn) The method according to claim 64, wherein said event being generated comprises a pause.
- Claim 80 (withdrawn) The method according to claim 64, wherein said event being generated comprises a start-up.
- Claim 81 (withdrawn) The method according to claim 64, wherein said event being generated comprises an audit event.
- Claim 82 (withdrawn) The method according to claim 81, wherein said audit event is an add audit command.
- Claim 83 (withdrawn) The method according to claim 81, wherein said audit event is a modify audit command.

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

Claim 84 (withdrawn) The method according to claim 81, wherein said audit event is a stop audit command.

Claim 85 (withdrawn) The method according to claim 64, wherein said event being generated comprises use of extended stored procedures.

Claim 86 (withdrawn) A method for providing exceptions to security alerts, the method comprising:

monitoring for events generated by a database application;

filtering alerts raised that match a defined set of rules;

passing alerts not matching a normal definition of said defined set of rules.

Claim 87 (withdrawn) The method according to claim 86, wherein said defined set of rules comprises values for each field collected for each event.

Claim 88 (withdrawn) The method according to claim 86, wherein said filtering is matched by comparing values of each field with values defined in an exception.

Claim 89 – 97 (cancelled)

Claim 98 (currently amended) A computer readable medium having code to perform a computer implemented method for protecting a database hosted on a server, comprising:

installing a console on a remote computer system for monitoring activity on the database, the remote computer system having a first tangible computer readable medium;

presenting the installed console through a user interface;

the user interface being displayed on a monitor;

registering a listener agent with the console; the listener agent being installed on the server hosting the database, the server having a second tangible computer readable medium;

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

establishing a secure connection between the console and the listener agent
wherein the secure connection operates over a secure socket layer and the console and the
listener agent are in cross-platform communication using simple object access protocol;

the console and the listening agent monitoring activity at an application level of
the database;

configuring the listener agent with a first set of rules having a set of security
attributes;

installing a collector agent to be in communication with the listener agent for
collecting a plurality of database events wherein the collector agent includes a plurality of
collector definitions, each one of the collector definitions being associated with a database
instance;

deconstructing the plurality of database events into a plurality of atomic
messages;

analyzing the plurality of atomic messages for compliance with the first set of
rules;

executing compliant database events;

transmitting a signal to a console operator when a database event is not compliant
with the first set of rules, wherein transmitting the signal to the console includes using a
dispatcher agent connected to the console over a peer-to-peer channel and transmission of the
signal is platform independent;

allowing a console operator to create exceptions to the first set of rules when
signals are sent by the listener agent;

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

updating the first set of rules with the exceptions created by the console operator;
storing the signals received by the console operator in a data file residing with the console, in association with the second tangible computer readable medium.

Claim 99 (previously presented) The computer readable medium having code to perform a computer implemented method for protecting a database hosted of Claim 98, wherein the step of analyzing further comprises the steps of:

determining whether the plurality of atomic database events include an executable SQL statement that exploits a buffer overflow vulnerability in the database;
preventing the executable SQL statement from executing.

Claim 100 (previously presented) The computer readable medium having code to perform a computer implemented method for protecting a database hosted of Claim 98, wherein the step of analyzing further comprises the steps of:

detecting whether an executable SQL statement includes an operating system call;
preventing the executable SQL statement from making the operating system call.

Claim 101 (previously presented) The computer readable medium having code to perform a computer implemented method for protecting a database hosted of Claim 98, wherein the step of analyzing further comprises the steps of:

determining whether an executable SQL statement contains a write operation to a data dictionary;
preventing the data dictionary from being written to.

Appl. No. 10/798,079
Amdt. Dated September 16, 2010
Reply to Office action of June 18, 2010

Claim 102 (previously presented) The computer readable medium having code to perform a computer implemented method for protecting a database hosted of Claim 98, wherein the step of analyzing further comprises the steps of:

determining whether an executable SQL statement alters a set of auditing configurations existing on the database;

preventing the set of auditing configurations from being altered.

Claim 103 (previously presented) The computer readable medium having code to perform a computer implemented method for protecting a database hosted of Claim 98, wherein the step of analyzing further comprises the steps of:

determining whether an executable SQL statement includes a write operation to a set of audit records existing in a log file;

preventing the audit records existing in the log file from being written to.

Claim 104 (previously presented) The computer readable medium having code to perform a computer implemented method for protecting a database hosted of Claim 98, wherein the step of analyzing further comprises the steps of:

determining whether an executable SQL statement includes an attempt by a user to obtain administrator access by changing a configuration file in the database;

preventing the configuration file in the database from being changed.